Listing of Claims:

1. (Currently Amended) A buffer manager, comprising:

an input for receiving packets of data, each packet associated with an output queue;

an intermediate storage facility having a plurality of blocks; [[and]]

an intermediate storage facility manager configured to assign particular blocks of the intermediate storage facility to buffer packet data according to destination output queues, and store one or more packets associated with the output queues into the blocks assigned to those output queues;

a queue memory; and

a second storage facility manager coupled to the intermediate storage facility manager and configured to accept a command from the intermediate storage facility manager and, upon receipt of the command, store into the queue memory one or more groups of blocks that were previously stored in the intermediate storage facility, the second storage facility manager comprising a memory access controller coupled to the queue memory and a memory bank scheduler coupled to the memory access controller and configured to direct portions of groups of blocks to particular banks of the queue memory.

2. (Original) A buffer manager according to claim 1 wherein the intermediate storage facility manager comprises:

a pointer repository for tracking locations in the intermediate storage facility;

a trunk manager configured to interact with the pointer repository to store locations of trunks stored in the intermediate storage facility.

3. (Original) A buffer manager according to claim 2 wherein each trunk managed by the trunk manager is made of one or more blocks of the intermediate storage facility having a common output queue.

4. (Original) A buffer manager according to claim 2 wherein the intermediate storage facility manager further comprises a temporary storage memory circuit for storing the packets of data prior to the time that the packets of data are stored in the intermediate storage facility.

5. (Original) A buffer manager according to claim 1 wherein the intermediate storage facility is an SRAM circuit.

6. (Previously Presented) A buffer manager according to claim 1 wherein the output queues are virtual output queues sharing a common queue memory.

7. (Canceled).

8. (Canceled).

9. (Currently Amended) A buffer manager according to claim [[8]] 1, wherein the second storage facility manager comprises a dynamic balancer, including:

a token register including a number of tokens,

a token distributor configured to allocate the number of tokens between read and write process managers,

the write process manager configured to accept a number of write tokens from the token distributor and authorize a number of data write operations to the queue memory equal to the number of write tokens received, and

the read process manager configured to accept a number of read tokens from the token distributor and authorize a number of data read operations to the queue memory equal to the number of read tokens received.

10. (Original) A buffer manager according to claim 9 wherein the dynamic balancer is configured to force all of the number of data write operations to be performed before any of the data read operations are performed.

11. (Currently Amended) A buffer manager according to claim [[7]] 1, further comprising an output queue manager configured to supervise output queues made of one or more trunks.

12. (Currently Amended) A buffer manager according to claim [[7]] 1 wherein the queue memory is an SDRAM circuit.

13. (Currently Amended) A buffer manager according to claim [[7]] 1, further comprising a third storage facility coupled to the second storage facility manager, the third storage facility able to store groups of blocks that were previously stored in the queue memory.

14. (Currently Amended) A buffer manager according to claim [[7]] 1, further comprising:

an output SDRAM storage circuit;

an output SDRAM controller coupled to the second storage facility manager, the output SDRAM controller circuit configured to store trunks of data that were previously stored in the queue memory.

15. (Currently Amended) A line interface card, comprising:

one or more input ports configured to receive packets;

one or more output ports configured to transmit packets; and

a packet buffer manager, including

a buffer memory having blocks of storage locations;

a buffer memory manager configured to sort the data packets into groups and store the groups into the buffer memory blocks, wherein the data packets are sorted into groups having common assigned output queues for storage into the buffer memory blocks;

a block storage memory comprising the output queues to receive the grouped packet data from the buffer memory; and

a block storage memory manager coupled to the buffer memory manager, comprising a memory access controller coupled to the block storage memory and a memory bank scheduler coupled to the memory access controller and structured to direct portions of the one or more groups of blocks to particular banks of the block storage memory.

18. (Canceled).

19. (Previously Presented) A line interface card according to claim 15, further comprising a second block storage memory coupled to the buffer memory manager, the second block storage memory able to store groups of blocks that were previously stored in the block storage memory.

20. (Previously Presented) A line interface card according to claim 15, further comprising:

an output SDRAM storage circuit;

an output SDRAM controller coupled to the block storage memory manager, the output SDRAM controller circuit structured to store trunks of data that were previously stored in the block storage memory.

21. (Original) A line interface card according to claim 15 wherein the computer network is the Internet.

22. (Original) A line interface card according to claim 15 wherein the buffer memory manager further comprises a temporary storage memory circuit for storing the packets of data prior to the time that the packets of data are stored in the buffer memory.

23. (Original) A line interface card according to claim 15, further comprising a packet processor coupled to one or more of the input ports and output ports.

24. (Currently Amended) A network device, comprising:

one or more input ports;

one or more output ports;

a switching fabric connecting selected input ports to selected output ports;

a packet buffer manager, including

a buffer memory having a plurality of storage location blocks,

a buffer memory manager configured to sort data packets accepted from the input ports into groups and store the groups into one or more of the plurality of the storage location blocks in the buffer memory, wherein the data packets are sorted into groups having common assigned output queues for storage into the buffer memory blocks,

a block storage memory comprising the output queues to receive the grouped packet data from the buffer memory, and

a block storage memory manager coupled to the buffer memory manager, comprising a memory access controller coupled to the block storage memory and a memory bank

scheduler coupled to the memory access controller and configured to direct portions of groups of blocks to particular banks of the block storage memory; and

a scheduler configured to direct the packet buffer manager to output the groups through the switching fabric.

27. (Previously Presented) A network device according to claim 24 wherein the block storage memory manager is configured to accept a command from the buffer memory manager and, upon receipt of the command, store into the block storage memory one or more groups of blocks that were previously stored in the buffer memory.

28. (Canceled).

29. (Previously Presented) A network device according to claim 24, further comprising a second block storage memory coupled to the buffer memory manager, the second block storage memory able to store groups of blocks that were previously stored in the block storage memory.

30. (Previously Presented) A network device according to claim 24, further comprising:

an output SDRAM storage circuit;

an output SDRAM controller coupled to the block storage memory manager, the output SDRAM controller circuit structured to store trunks of data that were previously stored in the block storage memory.

31. (Original) A network device according to claim 24 wherein the computer network is the Internet.

32. (Currently Amended) A network device, comprising:

one or more input ports structured to accept data packets from a computer network, each packet having an assigned output queue;

one or more output ports structured to send data packets onto the computer network;

a switching fabric coupled to the one or more input ports and the one or more output ports and structured to connect selected input ports to selected output ports;

a packet buffer manager, including

an input coupled to the one or more input ports,

a buffer memory having a plurality of storage location blocks, each block able to store at least a portion of the data packets accepted from the one or more input ports,

a buffer memory manager structured to sort the data packets accepted from the one or more input ports into groups, store the groups into one or more of the plurality of the storage location blocks in the buffer memory, and retrieve one or more of the stored groups, wherein the data packets are sorted into groups having common assigned output queues for storage into the buffer memory blocks,

a block storage memory comprising the output queues to receive the grouped packet data from the buffer memory, and

a block storage memory manager coupled to the buffer memory manager, comprising a memory access controller coupled to the block storage memory and a memory bank scheduler coupled to the memory access controller and configured to direct portions of groups of blocks to particular banks of the block storage memory; and

a scheduler coupled to the packet buffer manager and to the switching fabric, the scheduler structured to direct the packet buffer manager to read one or more of the stored groups and to direct the groups read from the buffer memory through the switching fabric.

35. (Currently Amended) A method for buffering packet data in a network device, comprising:

receiving data packets at an input port, each data packet having a predetermined output queue;

aligning the data packets into groups of packet data, each group comprising packet data having the same output queue, wherein aligning the data packets into packet data groups each having the same output queue comprises

determining a required number of blocks in the memory buffer to store the packet data having the same output queue,

obtaining addresses of the required number of free blocks in the memory buffer, and

creating an ordered list of the addresses obtained;

buffering the packet data groups in the blocks of a memory buffer arranged by blocks; and

selecting buffered packet data groups for storage in a memory device comprising the output queues.

36. (Previously Presented) A method for buffering packet data according to claim 35, further including creating a list of the blocks used to store the grouped packet data having the same output queues.

37. (Previously Presented) A method for buffering packet data according to claim 35, further comprising, upon receiving a signal:

reading a selected packet data group previously stored in the blocks of the memory buffer; and

removing the read grouped data packets from the memory buffer.

38. (Previously Presented) A method for buffering packet data according to claim 37 that further comprises storing the selected packet data group in the memory device after the selected packet data group has been read from the memory buffer.

39. (Canceled).

40. (Currently Amended) ~~A method for buffering packet data according to claim 35~~ A method for buffering packet data in a network device, comprising:

receiving data packets at an input port, each data packet having a predetermined output queue;

aligning the data packets into groups of packet data, each group comprising packet data having the same output queue, wherein aligning the data packets into packet data groups each having the same output queue comprises[[:]]

determining a required number of blocks in the memory buffer to store the packet data having the same output queue[[;]],

requesting pointers to the required number of free blocks in the memory buffer from a free block pool[[;]], and

creating a linked list of the pointers obtained from the free block pool[[.]];

buffering the packet data groups in the blocks of a memory buffer arranged by blocks; and

selecting buffered packet data groups for storage in a memory device comprising the output queues.

41. (Currently Amended) A method for buffering packet data according to claim [[35]] 40, further comprising:

receiving an additional packet at the input port, the additional packet having the same output queue as a packet data group currently stored in the memory buffer; and

storing packet data from the additional packet in the packet data group in the memory buffer and having the same output queue.

42. (Previously Presented) A method for buffering packet data according to claim 41 wherein storing packet data from the additional packet in the packet data group comprises adding additional pointers to a linked list representing the blocks in the memory buffer assigned to the packet data group.

43. (Previously Presented) A method for buffering packet data according to claim 38 wherein storing the selected packet data group in the memory device comprises storing the selected packet data group in an SDRAM memory device as a single unit.

44. (Previously Presented) A method for buffering packet data according to claim 37 wherein the signal is generated when one of the following conditions is true:

when a pre-set time after storing the selected packet data group in blocks of the memory buffer has elapsed; or

when an amount of data stored in the blocks of the memory buffer equals or exceeds a threshold.